

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Pintar, et al.
Serial No.: 09/273,149
Filing Date: March 19, 1999
Confirmation No.: 6715
Group Art Unit: 2178
Examiner: Paula, Cesar B.
Title: **SYSTEM FOR GENERATING OPTIMIZED
COMPUTER DATA FIELD CONVERSION
ROUTINES**

MAIL STOP AMENDMENT

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

DECLARATION PURSUANT TO 37 C.F.R. § 1.131

I, the undersigned, hereby declare and state that:

1. I am over the age of 21 years, of sound mind, and competent in all respects to make this Declaration.
2. I am an inventor of the subject matter of the above-referenced patent application, entitled *System for Generating Optimized Computer Data Field Conversion Routines*, filed on March 19, 1999 (the "Application").
3. Prior to March 16, 1999, I gained a full understanding of the subject matter of at least the current version of Claims 1-15 and 18-20 and proposed new Claims 21-26 of the Application (the "Invention"), as provided to me in a draft amendment by Baker Botts,

L.L.P., the draft amendment attached hereto as Appendix A; therefore, I conceived the subject matter of the Invention prior to March 16, 1999.

4. Beginning after conception of the Invention, I participated in the design and creation of a software program that incorporated the subject matter of the Invention (the "Software") and that was fully operational prior to March 16, 1999; therefore, I reduced the subject matter of the Invention to practice prior to March 16, 1999.

5. All of the work in conceiving and reducing the Invention to practice occurred in the United States. The computers used to create and run the software programs resided within the United States.

6. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true. Further, I declare that these statements are made with the knowledge that willful false statements, and the like so made, are punishable by fine or imprisonment, or both, under Section 1001, Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the Application or any patent issuing thereon.

[THE REST OF THE PAGE INTENTIONALLY LEFT BLANK]

Declaration pursuant to 37 C.F.R. § 1.131 in regard to 09/273,149.

Signed this 20 day of June, 2006.

A handwritten signature in black ink, appearing to read "Kevin M. Pinter", is written over a horizontal line.

Kevin M. Pinter

EXHIBIT A
DRAFT AMENDMENT OF THE CLAIMS

1. (Previously Presented) A method of converting data of a plurality of input data types to a plurality of output data types by an application program, said method comprising:
 - (a) receiving a first attribute of a first input data type and a second attribute of a first output data type;
 - (b) dynamically creating at runtime a first optimized conversion routine based on said first attribute and said second attribute, the conversion routine including one or more computer instructions to be executed during conversion;
 - (c) validating specific field conversion options of the conversion routine; and
 - (d) executing said first optimized conversion routine from said application program to convert data of said first input data type to said first output data type.
2. (Previously Presented) The method of claim 1, wherein step (d) comprises calling said first optimized conversion routine from said application.
3. (Previously Presented) The method of claim 1, wherein step (d) comprises storing said first optimized conversion routine inline with said application.
4. (Original) The method of claim 1, wherein step (b) is performed dynamically while said application program is executing.
5. (Previously Presented) The method of claim 1, further comprising:
 - (e) receiving a third attribute of a second input data type and a fourth attribute of a second output data type;
 - (f) generating a second optimized conversion routine based on said third attribute and said fourth attribute; and
 - (g) executing said second optimized conversion routine from said application program to convert input data of said second input data type to said second output data type.

6. (Original) The method of claim 1, wherein said first and second attribute is character type.

7. (Original) The method of claim 1, further comprising generating program debugging instrumentation for said first optimized conversion routine.

8. (Currently amended) A method of converting data from input data types to output data types, said method comprising:

(a) receiving a plurality of sets of input attributes and output attributes from an application program, each set comprising one or more input attributes and one or more output attributes, the input attributes in a set associated with a first data type and the output attributes in the set associated with a second data type different than the first data type;

(b) dynamically creating at runtime a data conversion routine for each set of input attributes and output attributes, each conversion routine configured to convert data of the first data type of the associated set to the second data type of the associated set, each conversion routine including one or more computer instructions to be executed during conversion; and

(c) storing each data conversion routine in memory accessible to said application program.

9. (Currently Amended) The method of Claim 8, further comprising determining the size of the data conversion routine for each set of input attributes and output attributes.

10. (Currently Amended) The method of Claim 8, further comprising determining whether the data conversion routine for each set of input attributes and output attributes should be callable by said application program or should be stored inline with said application program.

11. (Previously Presented) The method of claim 8, wherein step (b) is performed dynamically while said application program is executing.

12. (Original) The method of claim 8, wherein said input and output attributes are character type.

13. (Original) The method of claim 8, wherein said input and output attributes are date type.

14. (Currently amended) The method of claim 8, further comprising generating program debugging instrumentation for each data conversion routine.

15. (Currently Amended) A system for dynamically generating computer data conversion routines, said system comprising:

a processor; and

a memory device coupled to said processor;

wherein said system is adapted to receive a plurality of sets of input attributes and output attributes from an application program, each set comprising one or more input attributes and one or more output attributes, the input attributes in a set associated with a first data type and the output attributes in the set associated with a second data type different than the first data type; and

wherein said memory device stores instructions that, when executed by said processor, cause said processor to:

dynamically create at runtime a data conversion routine for each set of input attributes and output attributes, each conversion routine configured to convert data of the first data type of the associated set to the second data type of the associated set, each conversion routine including one or more computer instructions to be executed during conversion; and

store each data conversion routine in a memory device accessible to said application program.

16. (Canceled)

17. (Canceled)

18. (Currently amended) The system of claim 15, wherein each data conversion routine is created while said application program is executing.

19. (Original) The system of claim 15, wherein said input attributes are character type and said output attributes are date type.

20. (Currently amended) The system of claim 15, wherein said memory device stores further instructions that, when executed by said processor, cause said processor to generate program debugging instrumentation for each data conversion routine.

21. (New) The system of claim 15, wherein said memory device stores further instructions that, when executed by said processor, cause said processor to determine the size of the data conversion routine for each set of input attributes and output attributes.

22. (New) The system of claim 15, wherein said memory device stores further instructions that, when executed by said processor, cause said processor to determine whether the data conversion routine for each set of input attributes and output attributes should be callable by said application program or should be stored inline with said application program.

23. (New) Logic encoded in a computer-readable medium, the logic operable when executed by a computer to:

(a) receive from an application program at least a first attribute associated with a first input data type and a second attribute associated with a first output data type;

(b) while the application program is executing, dynamically create a first conversion routine based on the first attribute and the second attribute, the first conversion routine configured to convert data of the first input data type to the first output data type, the first conversion routine including one or more computer instructions to be executed during conversion; and

(c) store the first data conversion routine in a memory device accessible to the application program.

24. (New) The logic of claim 23, wherein the first data conversion routine is stored such that the application program calls the first data conversion routine when executing the first data conversion routine.

25. (New) The logic of claim 23, wherein the first data conversion routine is stored inline with the application program.

26. (New) The logic of claim 23, further operable when executed to generate program debugging instrumentation for the first data conversion routine.